Customer Story

Adlib

From 1 year to 1 month: Turbocharging digital document conversion with Adlib

Challenge

McKesson Corporation was facing a discontinuation of the software application they had been using to convert and upload data into their SharePoint system, and looking for a solution that offered long-term file accessibility.

Solution

McKesson implemented a new document management system, using Adlib for PDF document conversion and merging.

Results

A highly automated process for managing/converting large volumes of documents to PDF/A

- Guaranteed file longevity on critical documents
- PDF/A conversion/validation capabilities enable longterm archiving
- Form backlog was cleared and 11 months was saved in conversion process

Overview

McKesson Corporation is a Fortune 500 company distributing pharmaceuticals and providing health information technology, medical supplies, and care management tools. The company needed a new solution to reduce the manual work and increase speed in converting and merging documents for upload into SharePoint, while ensuring long-term file accessibility, and fully documented versioning of forms.

Challenge

My team of four and I are essentially the gatekeepers for any software about to be released from McKesson Corporation. We are responsible for product release requirements, and any software we're selling has to come through our team to ensure it meets our policies.

Originally, we were using spreadsheets and emails to gather product release requirements. But since I had knowledge of Microsoft InfoPath, I was able to build a form to make this process more efficient. My form helps us to interact with everyone at the company. We receive the internal policies from the executive team and figure out how we need to collect information to ensure the different business units are meeting those policies. The form has since been rebuilt in K2 and with attachments can be anywhere from ten pages to a thousand pages.

Based upon which fields have been filled out, work list items are triggered so our team can go in and verify everything's been done correctly. At the end of it, we have the finalized, completed form, which then lets our Ireland team know they can go ahead and distribute the software. But then, we need to keep these records for 12 years.

Protecting the Past, Stepping into the Future

Keeping digital records for a long period of time becomes an issue when you realize how quickly technology changes. I knew there was no way we'd be able to open the form I created 10 years down the road and see what it looked like when it was created. On top of that, in 2014 Microsoft announced they would discontinue InfoPath. We were forced to look for an alternative and, ideally, better solution.

Solution

Since I managed the form, the SharePoint site, and all the records management, I needed a simple-to-use solution. We also had to deal with the form in InfoPath along with multiple attachments that were being saved to a SharePoint library. We wanted to be able to combine that form and all the attachments into one PDF, have it in the order that the attachments are seen on the form, and then have the final version be the PDF/A.

There were two reasons for PDF/A requirement. The first was records retention—the ability to prove exactly what the form looked like when something was approved. And secondly, policies change every year. The form had to change and we needed to make sure we were capturing what the form looked like when it was approved. This way, I could have one, unified library and a single form.

With PDF/A, it's very controlled. Its purpose is that when you open a PDF in 50 years, it looks exactly the same as when it was created, despite an infinite number of software updates. Since this was one of our main criteria in a new solution, I searched "PDF/A conversion" online and found about 20 solutions.

In typical librarian fashion, I documented the potential solutions in an Excel spreadsheet and cross-referenced their offerings with our requirements. A big one was our need to convert InfoPath and multiple files from the SharePoint attachment library.

Eventually, I narrowed it down to two tools that could do everything we needed: Muhimbi and Adlib. The final decision hinged on the fact that Muhimbi runs on the SharePoint site. This meant every time we triggered a conversion, it ran in the SharePoint environment, which would slow it down for other teams. We're a huge company with thousands of SharePoint sites and tens of thousands of people using SharePoint every day, so Muhimbi clearly wasn't the right solution for us. Adlib has its own separate environment where it captures the data, forms, and attachments, pulls them into its server, and does all the work there. That's what made it exciting for the SharePoint team and made us ultimately decide to go with Adlib.

Results

Getting-and Staying-on Track

McKesson has unique needs due to the nature of our policy requirements. Thankfully, Adlib accommodated our needs and created specialized rules as they've done in the past for other clients. We did a proof of concept with Adlib and had to make a few adjustments as we went along, but the platform did exactly what we needed. We had different tracks on our InfoPath form and, depending on the type of software we're building, we would have different tabs that may or may not appear. With Adlib, we were able to define which tabs our end-users needed to capture based on the specific track.

Before Adlib, it would've taken me about thirty minutes to convert a small form in Adobe Acrobat Pro, and between two or three hours for a big form. One of our attachments could be a .zip file of literally 50 or 60 approval emails. Now, with Adlib, what used to take me hours has been reduced to just a couple minutes.

Today, we convert about 20 forms per month, but when we started, there was a major backlog as we developed our new processes and team training. By the time we could truly concentrate on converting the forms again, I had about 250 forms I had to convert to get caught up. But since we had Adlib, I wasn't worried.

> What took me a month to do using Adlibprobably would've taken me a year without it.

Since we've integrated Adlib, I'm now able to focus on my other responsibilities while making improvements to the form. I can play with the form and let our teams know what the tool will allow us to do. I was also able to catch up on assigning retention codes to all of our SharePoint sites. On top of that, I could spend time setting up records retention schedules and requirements for a whole site collection. Essentially, using Adlib has helped me get back to doing more impactful work.

We're now able to quickly and easily get the rchival version of any form. With Adlib, we know exactly what the form and attachments looked like and the day it was approved. It allows me to create updates to a two-year-old form so I don't have to worry about people looking at an outdated form—not realizing certain fields were not requirements at that time. We can look up what our policy said at the time and see, very clearly, if we followed it.

> We've completely done away with any confusion when it comes to cross referencing documents to policies.

In addition to the work Adlib has enabled us to do, they've also been incredible to work with. Any time Adlib comes out with a new version, we get on the phone with them and they walk us through it. We do everything in QA, work out any kinks, and then we go to production. It's a non-event.

It's also incredibly easy to schedule a WebEx meeting so they can see how the platform is working in our environment. And when we switched to K2, they helped us immensely with the integration. Adlib has been absolutely fantastic—one of the best third-party vendors I've worked with.

In addition to the work Adlib has enabled us to do, they've also been incredible to work with. Any time Adlib comes out with a new version, we get on the phone with them and they walk us through it. We do everything in QA, work out any kinks, and then we go to production. It's a non-event. It's also incredibly easy to schedule a WebEx meeting so they can see how the platform is working in our environment. And when we switched to K2, they helped us immensely with the integration. Adlib has been absolutely fantastic—one of the best third-party vendors I've worked with.

Building a Clear Vision

What helped us most with our implementation is that we determined our desired results, and then worked backwards. We knew exactly what we required from Adlib. We needed to capture the product name and release version for our headers; capture the PRR number; and pull data from the InfoPath library and from an attachment library. Collecting all those requirements up front helped tremendously. Then, we met with the Adlib team to show them exactly what we needed and found the best solution in the shortest amount of time.

If I hadn't done all that pre-work, we still would've gotten there, but it would've been a lot more trial and error. With any major project like this, it's important to ask the relevant questions ahead of time, organizing your systems, and partner with the right vendor to set yourself up for success. Finally, my fellow software gatekeepers and I can focus on the core tasks that matter to us most.